

CLAIMS

What is claimed is:

1. A model-based management system for managing an application or service, comprising:
 - a description component that describes the application or service in terms of its constituent components, and desired states in terms of at least one of functionality, configuration, system resource utilization, security, and performance; and
 - a management service component that uses the description component during installation of the application or service to configure itself.
2. The system of claim 1, the management service component ensures availability of the application through management actions that include at least one of configuration management, problem detection, diagnosis, and recovery.
3. The system of claim 1, the description component comprises a models component that models one or more of the constituent components, health states and recovery, configuration settings, and administrative tasks.
4. The system of claim 1, the description component comprises a manifest component that contains information associated with models and source code attribution in a machine-readable form for use by the management service.
5. The system of claim 1, the description component comprises a management system component that comprises multiple services configured from information received from an application manifest.
6. The system of claim 1, the description component comprises an administrative tasks component that includes administrative tasks defined in an application manifest.

7. The system of claim 1, the description component comprises an attribution component that facilitates inserting attribution data in source code to indicate instrumentation and logic for monitoring aspects of the application.

8. The system of claim 1, the description component comprises a management system component that uses the desired states expressed in a manifest.

9. The system of claim 8, the management system component uses the desired states as modified by an administrator.

10. The system of claim 9, the desired states verify dependencies and install only the necessary files, settings, and security data.

11. The system of claim 9, one or more of the desired states subscribe to events and forward the events according to a predetermined specification.

12. The system of claim 9, one or more of the desired states periodically collect at least one of instrumentation data and counter data.

13. The system of claim 9, one or more of the desired states perform automatic management tasks.

14. The system of claim 9, one or more of the desired states restrict access to program functions.

15. The system of claim 9, one or more of the desired states perform at least one of detecting problems, diagnosing root causes, taking corrective action, and notifying an administrator when intervention is necessary.

16. The system of claim 9, one or more of the desired states customize policy for use with a plurality of different computers.

17. The system of claim 1, further comprising a rules definition language (RDL) that enables defining a rule for monitoring availability of software and hardware components, the RDL facilitates at least one of problem testing, diagnosis, resolution, verification, and notification.

18. The system of claim 1, further comprising a uniform resource identifier (URI) employed to uniquely identify at least one of an abstract resource, a physical resource, and a collection of resources.

19. The system of claim 1, further comprising a URI template, which URI template allows a probe to be identified and characteristics of the probe understood without retrieving the probe.

20. The system of claim 1, further comprising an instrumentation catalog that utilizes a URI template to describe instrumentation without referring to a specific instance.

21. The system of claim 1, further comprising an attribution component that facilitates attribution of source code for monitoring health of the application.

22. The system of claim 1, further comprising an attribution component that facilitates determining which parts of source code is used to determine and/or correct health, and when to execute monitoring rules.

23. A computer system according to claim 1.

24. A system of model-based management, comprising:
- a description component that describes an application, service, and/or system in terms of constituent components and desired states, the constituent components include at least one of,
 - a model component that further includes at least one of a component model, a health model, a configuration model, an administrative task model, an architecture model, a performance model, and a security model;
 - a manifest component generated from at least one of the model components that includes constituent component information and attribution of source code of one of the model components;
 - a management system component that includes one or more application program interfaces (APIs) that interface with the application, service or system; and
 - a tasks component that defines at least one of monitoring tasks, troubleshooting tasks, and administrative tasks for performance by the model-based management system;
- and
- a management service component that uses the description component for deploying the application, service and/or system.

25. The system of claim 24, the APIs facilitate at least one of central configuration, role-based access, system monitoring, manifest storing and editing, event generation and logging, instrumentation, performance counter processing, local configuration, installation, automation, and task scheduling.

26. A computer-readable medium having computer-executable instructions that embodies the system of claim 24.

27. A method of model-based management for managing an application, comprising:

developing one or more models corresponding to components of the application, using a source code;

performing attribution of the source code to indicate what models or portions thereof will be monitored;

generating a manifest of manifest information corresponding to the modeled application components and source code attribution, the manifest information for use by a management system service;

configuring a plurality of the management system services based on the manifest information; and

expressing desired states in the manifest.

28. The method of claim 27, further comprising verifying dependencies and installing only at least one of necessary files, settings and security based on one or more of the desired states.

29. The method of claim 27, further comprising subscribing to events and forwarding the events according to a predetermined event specification based on one or more of the desired states.

30. The method of claim 27, further comprising polling instrumentation by periodically collecting instrumentation information, counter information, and tests based on one or more of the desired states.

31. The method of claim 27, further comprising performing automatic management tasks based on one or more of the desired states.

32. The method of claim 27, further comprising restricting access to program functions based on one or more of the desired states.

33. The method of claim 27, further comprising monitoring system processes based on the desired states by detecting problems, diagnosing root causes, taking corrective actions, and notifying the system administrator when intervention is required.

34. The method of claim 27, further comprising customizing policies and applying the customized policies to different computers to facilitate problem testing, diagnosis, resolution, verification, and notification.

35. The method of claim 27, further comprising:
determining one or more health states of a service;
publishing instrumentation of the application;
analyzing the published instrumentation; and
developing a health model of the service based on the published instrumentation, wherein the health model includes relationship information between components.

36. A model-based management system for managing an application or service, comprising:

means for describing the application or service in terms of its constituent components, and desired states in terms of at least one of functionality, configuration, security, and performance;

means for expressing management information along with source code of the application to facilitate determining health of the application;

means for identifying abstract or physical resources using a URI; and

means for configuring a management service component based in part on constituent components during installation of the application to configure the management service component.

37. The system of claim 36, the application or service is distributed.

38. A computer-readable medium having computer-executable instructions for performing a method of managing an application or service, the method comprising:

developing one or more models corresponding to components of the application, using a source code;

performing attribution of the source code to indicate what models or portions thereof will be monitored;

generating a manifest of manifest information corresponding to the modeled application components and source code attribution, the manifest information for use by a management system service;

configuring a plurality of the management system services based on the manifest information; and

expressing desired states in the manifest.

39. The computer-readable medium of claim 38, further comprising verifying dependencies and installing only at least one of necessary files, settings and security based on one or more of the desired states.

40. A computer-readable medium having computer-executable instructions that facilitates a model-based management system for managing an application or service, the system comprising:

a description component that describes the application or service in terms of its constituent components, and desired states in terms of at least one of functionality, configuration, system resource utilization, security, and performance; and

a management service component that uses the description component during installation of the application or service to configure itself.